Lab 2. Variables and Expressions. Flow of Control.

Introduction

So you have survived week 1 of INFO1103 and made it to the second lab, congratulations! There were many things that required setting in week 1: your University account, your PIAZZA account, assessing MyUni, eLearning and PIAZZA, installing Java and Eclipse on your home computer. There were some problems in the labs too, especially for the students who enrolled later or haven't used the SIT computers before (login and U drive access problems). Hopefully all problems and issues are resolved now, and we can focus on learning programming!

Lab sessions are very important in this course – they are where you practice concepts that are introduced and explained in the lectures. Computer programming is something you can only learn through practical experience, so the attendance at the labs is extremely important!

Each lab session consists of standard and challenge questions. The standard questions are compulsory. Note that sometimes we may not be able to finish all of the standard questions during the labs; you will need to do this at your own study time. The challenge questions are extensions of the standard questions; they are more difficult and are not compulsory. We will provide solutions to all questions every week (via eLearning, on Friday after the last lab).

The weekly homeworks start this week. Every week, except in weeks 7 and 13, you are required to complete a homework and submit it before Wednesday 9am in eLearning. The goal of the homework exercises is to prepare you better for the labs and also to encourage you to work steadily during the semester. Before completing the homework, go through the lecture slides and make sure that you understand them. Then do the homework, it is typically a straightforward application of the material covered in the lectures. You will benefit more from the labs if you come prepared, and this in turn will prepare you better for the assessments during the semester and the exam.

In the first few weeks we will provide the lab solutions in a pdf document, together with the lab questions. As we start writing bigger programs, we will change this to a zip archive and teach you how to import it in Eclipse. Warning: if you copy and paste code from a pdf document (e.g. the lab solutions) in Eclipse, there may be problems as pdf documents use different character encoding than Eclipse, and some symbols may not be readable in Eclipse, e.g. quotation marks, dashes, and even spaces. So please don't copy/paste but type your programs in Eclipse. This is also better from educational point of view at the beginning.

Homework

The homework for this week consists of 10 multiple choice questions organised as a test in eLearning. You need to answer them online, via eLearning. Login to eLearning, go to ***Course Material*** (left panel) and click on the link ***w2-homework*** to start the online test (homework). Important notes about the homeworks:

1. Do not delay the submission of the homework till the very last moment as eLearning may get overloaded and you may miss the deadline! You will not be able to submit after the deadline - the submission box will close at 9am exactly.

2. If you have problems submitting in eLearning before the deadline, email your answers to sit.info1103@sydney.edu.au (e.g. 1A, 2B, etc.).

3. You can submit twice, the last attempt will be marked.

Here are the questions for your information (remember that they must be answered online via eLearning!):

1. What will be the value of `a` and `b` after this code executes?

```
int a = 20;
int b = 10;
a = (a + b) / 2;
b = a;
a++;
```

A) a = 15, b = 16
B) a = 16, b = 16
C) a = 16, b = 15
D) a = 15, b = 15

2. What is wrong with the following code snippet?

```
int price;
price = 9.42;
```

A) The `price` variable is never initialized.
B) The data type for the `price` variable is not specified.
C) The `price` variable is never assigned a value.
D) The `price` variable is assigned a decimal value.

3. What will be the value of `var` after this code executes?

```
public static void main(String[] args)
{
   int var = 30;
   var = var + 2 / var;
   var++;
}
```

A) 0
B) 1
C) 30
D) 31

4. Which is the Java equivalent of the following mathematical expression?

$c = \sqrt{(a^2 + b^2)}$

A) `c = Math.sqrt(a * 2 + b * 2);`
B) `c = Math.sqrt(a * 2) + Math.sqrt(b * 2);`

C) `c = Math.sqrt(Math.pow(a, 2) + Math.pow(b, 2));`
D) `c = Math.sqrt(Math.pow(a, 2)) + Math.sqrt(Math.pow(b, 2));`

5. What will be printed?

```
String name = "Joanne Hunt";
System.out.println(name.length());
```

A) 8
B) 10
C) 9
D) 11

6. Assuming that the user enters "Joe" at the prompt, what is the output of the following code snippet?

```
public static void main(String[] args)
{
   System.out.print("Enter your name ");
   String name;
   Scanner in = new Scanner(System.in);
   name = in.next();
   name += ", Good morning";
   System.out.print(name);
}
```

A) The code snippet does not compile because the += operator cannot be used in this context.
B) Joe, Good morning
C) , Good morning
D) Joe

7. Assuming that the user provides 99 as input, what is the output of the following code snippet?

```
int a;
int b;
a = 0;
Scanner in = new Scanner(System.in);
System.out.print("Please enter a number: ");
b = in.nextInt();
if (b > 300)
{
   a = b;
}
else
{
   a = 0;
}
System.out.println("a: " + a);
```

A) `a: 0`
B) `a: 99`
C) `a: 100`
D) `a: 300`

8. What is the output of the following code snippet?

```
int s1 = 20;
if (s1 <= 20)
{
   System.out.print("1");
}
if (s1 <= 40)
{
   System.out.print("2");
}
if (s1 <= 20)
{
   System.out.print("3");
}
```

A) 1
B) 2
C) 3
D) 123

9. What is the output of the following code?

```
int age = 25;
if (age > 30)
{
   System.out.println("You are wise!");
}
else
{
   System.out.println("You have much to learn!");
}
```

A) There is no output due to compilation errors.
B) You are wise!
C) You have much to learn!
D) You are wise!
   You have much to learn!

10. In a `switch` statement, if a `break` statement is missing:

A) The `break` happens at the end of each branch by default
B) The statement will not compile
C) Execution falls through the next branch until a `break` statement is reached
D) The default case is automatically executed

1. Getting organised

Start Eclipse and create a package for this week's Java programming exercises:

- In the `Labs` project from last week, create a package called `lab2`
- For each exercise below create a class within this project and package

Please remember to create a package in `Labs` for your work every week. This will not be included in the lab notes any more but we will assume that you do it. Keeping your Java programs well organised is important!

2. Silly sentence

Write a program called `SillySentence.java` that asks the user to enter a name of a friend, a favourite color, food and animal, and then prints the following two line, with the user input replacing the items in italics:

```
I had a dream that name ate a color animal
and said it tasted like food!
```

For example, a possible dialog with the user is:

```
Enter the name of a friend:
Maria
Enter your favorite color:
blue
Enter your favorite food:
sushi
Enter your favorite animal:
dog
I had a dream that Maria ate a blue dog
and said it tasted like sushi!
```

3. Converting temperature from degrees Fahrenheit to degrees Celsius

Write a program called `FtoC.java` that converts a temperature from degrees Fahrenheit to degrees Celsius using the formula: degreesC = 5(degreesF-32)/9.

Prompt the user to enter a temperature in degrees Fahrenheit as a whole number without fractional part. Then have the program display the equivalent Celcius temperature, including the fractional part to two decimal points. A possible dialog with the user is:

```
Enter a temperature in degrees Fahrenheit (integer number):
72
72 degrees Fahrenheit = 22.22 degrees Celsius
```

4. Four-digit number

Write a program called `Vertical4Digits.java` that reads a four-digit integer, such as 1987, and then displays it, one digit per line like this:

```
1
9
8
7
```

Read the number as a string and use the method `chartAt` of the class `String` to get the four digits.

You need to include a message (prompt) telling the user to enter a four-digit integer. You can assume that that the user follows the instructions, i.e. there is no need to check if the number is not integer or not a four-digit integer.

5. Four-digit number – version two

The same task as above but this time read the number as an integer and use % to separate the four digits. Name the program `Vertical4Digits2.java.`

6. If statements

Consider the following code:

```
if (n > 10)
   System.out.print("*****");
if (n > 7)
   System.out.print("****");
if (n > 4)
   System.out.print("***");
if (n > 1)
   System.out.print("**");
System.out.println("*");
```

How many * will be printed when the code is executed

a) with n = 6 ?
b) with n = 20 ?
c) with n = 2 ?
d) with n = –1 ?

Note on the use of {} in the `if` branches: Recall that if we have more than one statement in an `if` branch, we need to use {} to group these statements together in a *compound statement*. If we have a single statement, as in the example above, we are not required to have {} but some programmers prefer always to use {} to avoid confusion. We recommend that you always do this. The program above is equivalent to this one:

```
if (n > 10)
{
   System.out.print("*****");
}
if (n > 7)
{
   System.out.print("****");
}
if (n > 4)
{
   System.out.print("***");
}
if (n > 1)
{
   System.out.print("**");
}
System.out.println("*");
```

7. Comparing strings

Copy and run the following program in Eclipse (i.e. you need to create a class `StringEqual` in the package for this week).

```
public class StringEqual {
         public static void main(String[] args){
            String str1 = "abcd";
            String str2 = "abcdefg";
            String str3 = str1 + "efg";
            System.out.println("str2 = " + str2);
            System.out.println("str3 = " + str3);
            if (str2 == str3)
               System.out.println("The strings are equal");
            else
               System.out.println("The strings are not equal");
         }
}
```

1) Does this program check if the two strings `str2` and `str3` are equal (i.e. have the same value)? If not, what does it compare?
2) Modify the program so that it compares the value of the two strings.

8. Paper, scissors, rock

Do you remember the game "Paper, scissors, rock"? It is a two-player game in which each person simultaneously chooses rock, paper or scissors. Rock beats scissors but loses to paper, paper beats rock but loses to scissors, and scissors beats paper but loses to rock. The following code prompts player 1 and player 2 to each enter a string: rock, paper, or scissors. Finish the code by adding `if` statements to appropriately report "Player 1 wins", "Player 2 wins" or "It is a tie." Run your code in Eclipse.

```
import java.util.Scanner;
public class RockPaperScissors {
     public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Player 1: Choose rock, scissors, or paper:");
        String player1 = scan.next().toLowerCase();
        System.out.println("Player 2: Choose rock, scissors, or paper:");
        String player2 = scan.next().toLowerCase();
        (your code goes here…)
    }
}
```

9. Challenge

Write a program called `CheckDate.java` that checks if the date entered by the user is a valid date. Display the date and a message that indicates whether the date is valid or not, and if not, why. Here are some possible dialogues with the user:

```
Please enter a date to be checked:
day:
29
month:
2
year:
2012
Date is 29:2:2012
```

```
It is a valid date.


Please enter a date to be checked:
day:
29
month:
2
year:
2013
Date is 29:2:2013
It is not a valid date.
The reason it is invalid: The day value is greater than 28 in February in a non-leap year.
```

```
Please enter a date to be checked:
day:
0
month:
3
year:
2013
Date is 0:3:2013
It is not a valid date.
The reason it is invalid: The day value is less than one.


Please enter a date to be checked:
day:
31
month:
6
year:
2010
Date is 31:6:2010
It is not a valid date.
The reason it is invalid: The day value is greater than 30 in a month with 30 days.
```

- A valid month has a value from 1 to 12.
- A valid day has a value from 1 to 30 in September, April, June and November, from 1 to 28 in February in a non-leap year, from 1 to 29 in February in a leap year, from 1 to 31 in all other months.
- A leap year is any year that is: 1) divisible by 4 but not divisible by 100 or 2) divisible by 400.

When the day is not valid your program needs to print one of the following messages:

```
The day value is less than one
The day value is greater than 29 in February in a leap year.
The day value is greater than 28 in February in a non-leap year.
The day value is greater than 31 in a month with 31 days.
The month value is not from 1 to 12.
```